

Transmission conditions with constraints in finite element domain decomposition methods for flow problems

G. Houzeaux and R. Codina*

Departament de Resistència de Materials i Estructures a l'Enginyeria, Campus Nord - Universitat Politècnica de Catalunya, Edifici C1, Jordi Girona 1-3, 08034 Barcelona, Spain

SUMMARY

This work presents a conservative scheme for iteration-by-subdomain domain decomposition (DD) strategies applied to the finite element solution of flow problems. The DD algorithm is based on the iterative update of the boundary conditions on the interfaces between the subregions, the so-called transmission conditions. The transmission conditions involve the essential and natural boundary conditions of the weak form of the problem, and should ensure strong continuity of the velocity and weak continuity of the traction. As a first approach, the transmission conditions are interpolated using the classical Lagrange interpolation functions. Conservation problems might arise when two adjacent subdomains have a sensibly different mesh spacing. In order to conserve any desired quantity of interest, an interface constraining is introduced: continuity of the transmission conditions are constrained under a scalar conservation equation. An example of mass conservation illustrates the algorithm. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: domain decomposition; conservation; finite elements; Stokes problem

1. GOVERNING EQUATIONS

1.1. The Stokes problem

Although our developments are applicable in a more general setting, for simplicity we will restrict ourselves to the Stokes problem

$$\begin{aligned} -2\mu\nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned}$$

where \mathbf{u} is the velocity field, p is the pressure, \mathbf{f} is the vector of body forces, $\boldsymbol{\varepsilon}(\mathbf{u})$ is the symmetrical part of the velocity gradient, and μ is the viscosity of the fluid. These equations

*Correspondence to: R. Codina, Departament de Resistència de Materials i Estructures a l' Enginyeria, Campus Nord - Universitat Politècnica de Catalunya, Edifici C1, Jordi Girona 1-3, 08034 Barcelona, Spain

Contract/grant sponsor: European Commission; contract/grant number: ESPRIT 23706

have to be solved in a domain Ω of dimension ns ($ns=2,3$), together with appropriate boundary conditions on the contour Γ . In particular, we will consider the conditions

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_g \quad \text{on } \Gamma_D \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{t}_n \quad \text{on } \Gamma_N \\ \mathbf{u} \cdot \mathbf{n} &= u_n, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}^1 = t_1, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}^2 = t_2 \quad \text{on } \Gamma_M \end{aligned}$$

where $\Gamma = \Gamma_N \cup \Gamma_D \cup \Gamma_M$, \mathbf{n} is the exterior normal, \mathbf{g}^1 and \mathbf{g}^2 are the local basis for the tangent space to Γ_M and $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$ is the Cauchy stress tensor. Note that for $ns=2$, only the prescription of t_1 is needed.

Let $V_{h,s}$ and $Q_{h,s}$ be appropriate finite element spaces to approximate the velocity and the pressure, respectively, and let $V_{h,t}$ and $Q_{h,t}$ be their test function counterparts. It is well known that $V_{h,s}$ and $Q_{h,s}$ have to satisfy the inf-sup or Babuška–Brezzi condition. In order to avoid this and, in particular, to be able to use equal velocity–pressure interpolations, we use the stabilized finite element formulation described for example in Reference [1], which can be viewed as a subgrid scale model [2]. It consists of finding $\mathbf{u}_h \in V_{h,s}$ and $p_h \in Q_{h,s}$ such that

$$\begin{aligned} 0 &= 2\mu \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u}_h) : \boldsymbol{\varepsilon}(\mathbf{v}_h) d\Omega - \int_{\Omega} p_h \nabla \cdot \mathbf{v}_h d\Omega + \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h d\Omega \\ &\quad - \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_h d\Omega - \int_{\Gamma_N} \mathbf{t}_n \cdot \mathbf{v}_h d\Gamma - \int_{\Gamma_M} (t_1 \mathbf{g}_1 + t_2 \mathbf{g}_2) \cdot \mathbf{v}_h d\Gamma \\ &\quad + \sum_{e=1}^{ne} \int_{\Omega_e} \tau (-\mu \Delta \mathbf{u}_h + \nabla p_h - \mathbf{f}) \cdot (\mu \Delta \mathbf{v}_h + \nabla q_h) d\Omega \end{aligned} \quad (1)$$

for the test functions $\mathbf{v}_h \in V_{h,t}$ and $q_h \in Q_{h,t}$. The first one only contains the Galerkin terms, whereas the last one introduces the stabilizing effect. The numerical parameter τ can be computed as

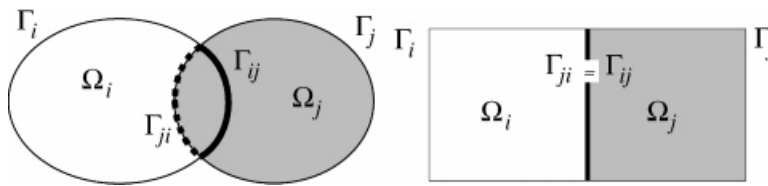
$$\tau = C \frac{h^2}{\mu}$$

where C is a constant that depends on the element type. For linear elements we take it as $C=1/4$, and for quadratics $C=1/16$.

2. THE DOMAIN DECOMPOSITION ALGORITHM

2.1. The coupling of the subdomains

The domain decomposition (DD) algorithm used in this work consists of solving a series of local problems linked through *transmission conditions*. This solution is performed in an iterative way, and thus ours is an *iteration-by-subdomain* DD formulation. Methods based on the iterative update of the boundary conditions are computationally very attractive. The implementation via a master-slave communication strategy enables the construction of the iterative process with very few alterations to the original flow solver. A master code controls the iterative process by performing the required operations (interpolation of the variables)

Figure 1. Two adjacent subdomains i and j .

and by passing the data (boundary conditions) back and forth to the slaves (the processes of the finite element code). A complete description of the numerical strategy we follow can be found in Reference [3]. For more information about domain decomposition methods see e.g. References [4, 5] and the references therein. In particular, for the Dirichlet/Neumann method see Reference [6]; for the Schwarz method see Reference [7]; for the Robin/Robin method see References [8, 9]; for adaptive methods see References [10, 11]; for domain decomposition methods as preconditioners see Reference [12]; for the mortar element method see Reference [13]; for the FETI method see Reference [14].

2.2. The transmission conditions

The transmission conditions are the essential and natural boundary conditions of the weak formulation of the governing equations; they ensure the continuity of the primary and secondary variables of the weak formulation. In the framework of the Stokes equations, they are the velocity and the traction components.

Let i and j denote two adjacent subdomains, possibly overlapping (see Figure 1). The velocity and traction continuities are expressed as

$$\begin{aligned} \mathbf{u}_i &= \mathbf{u}_j \\ \boldsymbol{\sigma}_i \cdot \mathbf{n}_i &= -\boldsymbol{\sigma}_j \cdot \mathbf{n}_j \end{aligned}$$

and should be imposed in some way on the interfaces. The first condition is a Dirichlet (D) condition. The second one is a Neumann (N) condition. The coupling of two subdomains i and j is performed by imposing one of these two conditions on the interface of i with j , namely Γ_{ij} , and another on the interface of j with i , namely Γ_{ji} . If the subdomains are disjoint, i.e. if $\Gamma_{ij} = \Gamma_{ji}$, the transmission conditions on both sides must be different. If the subdomains are overlapping, these two conditions can be the same. The possible methods are

- (i) a Dirichlet/Neumann coupling with disjoint subdomains,
- (ii) a Dirichlet/Dirichlet coupling with overlapping subdomains,
- (iii) a Dirichlet/Neumann coupling with overlapping subdomains.

The first method was first presented in the framework of the finite element method in Reference [15], and is extensively studied in Reference [4]. The second method is the classical Schwarz method, which was put back to the limelight in Reference [7]. The third one is presently under study by the authors, and is aimed to be applied to Chimera-type couplings [3].

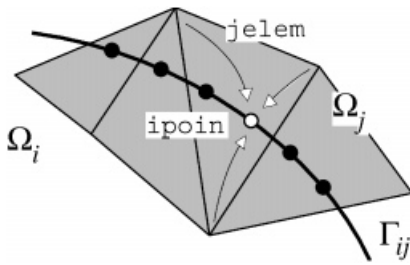


Figure 2. Classical interpolation from the underlying mesh.

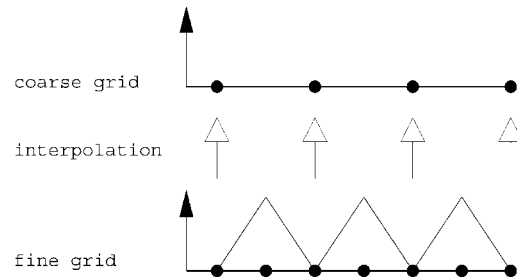


Figure 3. Interpolation from a fine grid to a coarse grid.

3. CONSERVATION

3.1. Classical interpolation

The update of the transmission conditions on the interface can simply be performed by interpolating the required unknown on the interface (involved in the interface boundary condition). Now, let us assume we want to update the interface transmission condition of subdomain i knowing the solution on subdomain j . The most straightforward strategy is the following (see Figure 2):

1. Loop over the interface nodes ipoin of i .
2. Find the host element jelem in the underlying subdomain j .
3. Interpolate the required transmission condition at ipoin from the solution in jelem using Lagrange interpolation functions.

This strategy is simple but non-conservative. Figure 3 illustrates the importance of using a conservative algorithm when interpolating a variable from a fine grid to a coarse grid. Although the continuity of the interpolated variable is assured on each node of the coarse grid, the global information is not necessarily well captured. High frequency modes may be filtered out if the variable exhibits strong variations along the interpolation domain. Several techniques are available to overcome the lack of conservation of classical interpolations; e.g. Cezal and Löhner [16] apply a weighted residual method to conserve the force when solving coupled fluid-structure problems. Different grid sizes are not the only reason for applying a conservative scheme; it can also be needed if the interpolated data are not compatible with the numerical formulation, as will be illustrated with the second numerical example.

The interface constraining presented in this work enables one to make a compromise between the continuity of the variable and the global information it carries. The general framework of the method will now be presented.

3.2. A conservative scheme

Let us assume we want to update the interface boundary condition of a scalar or vector variable a of subdomain i using the information of subdomain j . Denote by I^{ij} the continuous interpolation operator from mesh j to mesh i and by a_k the variable a referred to subdomain k ($k=i$ or j). As pointed out before, the idea of the interface constraining is to impose the

continuity of a variable via a classical interpolation and to relax it by the conservation of a global quantity. Therefore, we propose to find a_i from a_j by solving the following system:

$$\begin{aligned} &\text{minimize} && \int_{\Gamma_{ij}} |a_i - I^{ij}(a_j)|^2 d\Gamma \\ &\text{under the constraint} && f(a_i) = 0 \end{aligned}$$

where $f(a_i)$ is a function of the unknown a_i that determines the quantity to be conserved. For example, one can conserve the flux of a across the interface by choosing

$$f(a_i) = \int_{\Gamma_{ij}} \nabla a_i \cdot \mathbf{n}_i d\Gamma - \int_{\Gamma_{ij}} \nabla a_j \cdot \mathbf{n}_i d\Gamma$$

Let \mathbf{N} be the matrix of the classical Lagrange interpolation functions for the boundary elements of the interface and assume that f is an affine function. We can re-express the latter system in a matrix form as

$$\begin{aligned} &\text{minimize} && \int_{\Gamma_{ij}} |\mathbf{N}\mathbf{a}_i - \mathbf{N}\mathbf{I}^{ij}(\mathbf{a}_j)|^2 d\Gamma \\ &\text{under the constraint} && \mathbf{R}^t \mathbf{a}_i = r \end{aligned}$$

where \mathbf{R} and r are the vector and scalar representing the function f , respectively, \mathbf{a}_i is the nodal vector of a_i and \mathbf{I}^{ij} the discrete counterpart of I^{ij} . This system can be solved by introducing the Lagrange multiplier λ of the constraint. The Lagrangian is

$$L(\mathbf{a}_i, \lambda) = \int_{\Gamma_{ij}} |\mathbf{N}\mathbf{a}_i - \mathbf{N}\mathbf{I}^{ij}(\mathbf{a}_j)|^2 d\Gamma - \lambda(\mathbf{R}^t \mathbf{a}_i - r)$$

Register for free at <https://www.scipedia.com> to download the version without the watermark. Searching for the copyright of Scipedia, downloading the 2/2 version with the following system:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{R} \\ \mathbf{R}^t & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{a}_i \\ \mu_0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{I}^{ij}(\mathbf{a}_j) \\ r \end{bmatrix} \quad (2)$$

where

$$\mathbf{M} = \int_{\Gamma_{ij}} \mathbf{N}^T \mathbf{N} d\Gamma$$

Solving (2) for \mathbf{a}_i , we finally find

$$\mathbf{a}_i = \mathbf{I}^{ij}(\mathbf{a}_j) + (\mathbf{M}^{-1} \mathbf{R})(\mathbf{R}^t \mathbf{M}^{-1} \mathbf{R})^{-1}(r - \mathbf{R}^t \mathbf{I}^{ij}(\mathbf{a}_j))$$

Using a closed quadrature rule (for which the integration points are located on the nodes) to compute \mathbf{M} , this equation is trivial since the resulting approximation to \mathbf{M} is diagonal. Obviously, if instead of only one scalar constraint there are nc of them, exactly the same procedure can be applied. Matrix $\mathbf{R}^t \mathbf{M}^{-1} \mathbf{R}$ will then have $nc \times nc$ components.

In the next section, three examples will illustrate the method. The second numerical example presented will show that this conservation scheme not only enables to treat conservation problems due to different grids sizes but it can be necessary to conserve the mass when one of the subdomain is confined.

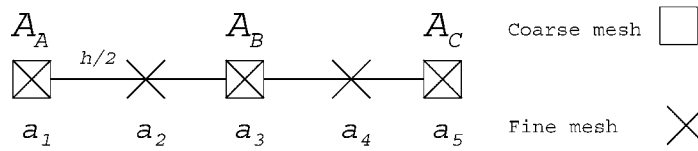


Figure 4. Example 1: coarse and fine meshes.

3.3. Example 1: analytical solution

We present a simple two-dimensional domain decomposition problem involving two disjoint subdomains. Figure 4 shows the nomenclature of both meshes on the interface, a one-dimensional line. The letters identify the coarse mesh nodes while the figures identify the fine mesh nodes; the capital letters refer to the coarse mesh solution while the small letters refer to the fine mesh solution. We propose to update the solution A of the coarse mesh, knowing the solution a of the fine mesh, imposing as a constraint the conservation of the integral of the solution. The corresponding problem is

$$\begin{aligned} & \text{minimize} \quad \int_0^{2h} |A - I(a)|^2 dx \\ & \text{under the constraint} \quad \int_0^{2h} A dx = \int_0^{2h} a dx \end{aligned} \quad (3)$$

We introduce A and a as the discrete vectors of unknowns of A and a , respectively. Using linear interpolation, the solution of the system is

$$A = I(a) + (M^{-1}R)(R^t M^{-1}R)^{-1}(r - R^t I(a))$$

Register for free at <https://www.scipedia.com> to download the version without the watermark

with

$$\begin{aligned} I(a) &= [a_1, a_3, a_5]^t \\ M &= \text{diag}(h/2, h, h/2) \\ R &= [h/2, h, h/2]^t \\ r &= h/2[a_1/2 + a_2 + a_3 + a_4 + a_1/2] \end{aligned}$$

All the integrals have been computed using a closed quadrature rule. Further calculations give

$$\begin{bmatrix} A_A \\ A_B \\ A_C \end{bmatrix} = \begin{bmatrix} a_1 \\ a_3 \\ a_5 \end{bmatrix} + \frac{1}{4} \left(-\frac{a_1}{2} + a_2 - a_3 + a_4 - \frac{a_5}{2} \right) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Now let us compare these results with those obtained with the classical interpolation, for the three triangle solutions shown in Figure 5. The solutions using the classical and constrained interpolations are drawn together with the fine mesh triangle solutions. Table I gives the results for the integration of the function along the interface. Obviously, only the constrained interpolation gives the right integral of the solution, the quantity conserved by solving

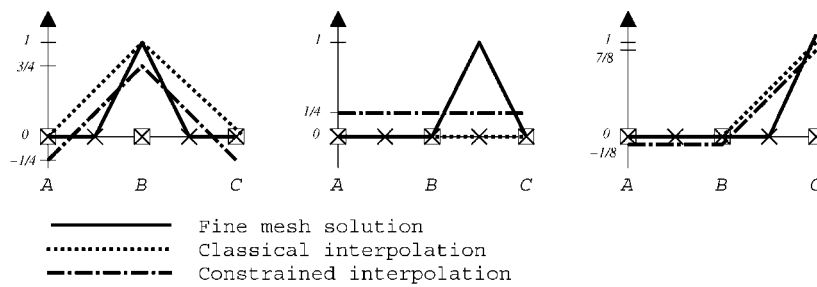


Figure 5. Example 1: interpolation of three triangle solutions.

Table I. Example 1: integration of triangle solutions.

Solution	Fine mesh	Classical	Constrained
$a_3 = 1$	$h/2$	h	$h/2$
$a_4 = 1$	$h/2$	0	$h/2$
$a_5 = 1$	$h/4$	$h/2$	$h/4$

system (3). Observe that in this case the nodal quadrature rule is exact, and therefore the integral of the unknown is exactly conserved.

In the next example, a similar constraint will be applied to the velocity field. We will see that under some numerical conditions, the conservation is necessary although the mesh nodes are at the interface to avoid the loss of accuracy of the solution.

3.4. Example 2: mass conservation

The second example presented involves mass conservation when one of the transmission conditions is of Dirichlet type and the Dirichlet subdomain is confined ($\Gamma_M = \Gamma_N = \emptyset$). Let us first tackle the origin of the need for conserving mass. The weak form of the continuity equation using the stabilized finite element method is derived taking $\mathbf{v}_h = 0$ in Equation (1):

$$\sum_{e=1}^{ne} \int_{\Omega_e} \tau \nabla q_h \cdot [-\mu \Delta \mathbf{u}_h + \nabla p_h] d\Omega + \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h d\Omega = 0$$

which must hold for all $q_h \in \mathcal{Q}_{h,i}$. Taking $q_h = 1$ in Ω , which is an admissible pressure test function, and integrating by parts, we obtain the following compatibility equation:

$$\oint_{\Gamma} \mathbf{u}_h \cdot \mathbf{n} d\Gamma = 0 \quad (4)$$

Consider two overlapping subdomains i and j . We want to update the interface Γ_{ij} boundary condition of i using a Dirichlet transmission condition. We assume Equation (4) is satisfied for subdomain j across the whole domain; however, zero mass flow rate across any interior section, and therefore across the interface Γ_{ij} , is not guaranteed. Therefore, we have

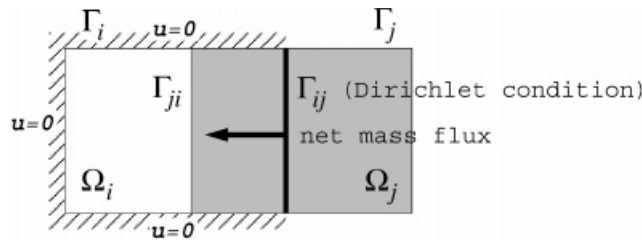


Figure 6. Example 2: a confined subdomain with Dirichlet condition on its interface.

generally that

$$\int_{\Gamma_{ij}} I^{ij}(\mathbf{u}_j) \cdot \mathbf{n}_i \, d\Gamma \neq 0 \quad (5)$$

Furthermore, if subdomain i is confined, the non-zero mass flow rate passing through the interface Γ_{ij} remains inside the subdomain and, therefore, the boundary condition of i does not verify the compatibility equation. This is illustrated by Figure 6. Note that for discontinuous pressure spaces, q_h can be taken piecewise constant and if Γ_{ij} coincides with some element boundaries of subdomain j , the net flux across Γ_{ij} is zero.

3.4.1. Constraining of the interface. One solution to circumvent the incompatibility of the transmission boundary condition and the weak formulation is to decrease the mass flux in subdomain i by means of constraining. We propose to obtain the boundary data \mathbf{u}_i by solving the problem

$$\begin{aligned} &\text{minimize} \quad \int_{\Gamma_{ij}} |\mathbf{u}_i - I^{ij}(\mathbf{u}_j)|^2 \, d\Gamma \\ &\text{under the constraint} \quad \oint_{\Gamma_i} \mathbf{u}_i \cdot \mathbf{n}_i \, d\Gamma = 0 \end{aligned}$$

The system is solved using the strategy defined in Section 3.2, with the constraint re-expressed as

$$\int_{\Gamma_{ij}} \mathbf{u}_i \cdot \mathbf{n}_i \, d\Gamma = - \int_{\Gamma_i \setminus \Gamma_{ij}} \mathbf{u}_i \cdot \mathbf{n}_i \, d\Gamma \quad (=0 \text{ in the case of Figure 6})$$

Note that the integral on the right-hand side of the constraint only involves the solution of i on $\Gamma_i \setminus \Gamma_{ij}$, which is known; the integral can therefore be calculated accurately using the same closed quadrature than that used to compute the boundary mass matrix \mathbf{M} . In the next example, a special integration rule will have to be designed as the right-hand side of the constraint will depend on the solution of the fine mesh.

3.4.2. Results. We solve the Stokes cavity flow using the $Q1/Q1$ element (piecewise bilinear velocities and pressures) on two subdomains. The DD method used to couple the subdomains is the Schwarz method. Figure 7 shows the pressure contours obtained.

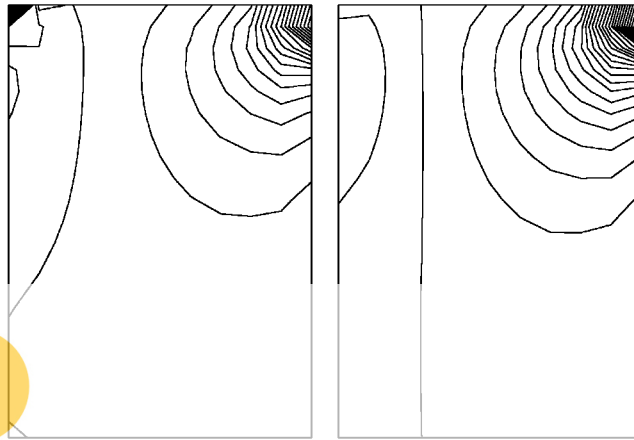


Figure 7. Example 2: pressure contours of the right-hand side subdomain. (Left) Classical interpolation. (Right) Constrained interpolation.

Subdomain	Classical	Constrained
Left-hand side	1.5	-0.1
Right-hand side	-3.0	0.1

The contours are only shown for the right-hand side subdomain. Figure 7(a) shows the contours only in the pressure field produced by the constrained method in mass is not in the subdomains. The wriggles appear precisely where the pressure is prescribed, i.e. at the top left corner. Figure 7(b) shows the pressure contours obtained using the zero mass flow rate constraint on the interface. The last solution corrects the zone of pressure instabilities in the upper left corner, where the value of the pressure is imposed (remember that the flow is confined and therefore, the pressure must be prescribed at one point in the subdomain). This is directly related to the fact that the algorithm enables to reduce the mass flow rate. This is confirmed by Table II which gives the net mass flow rates in the two parts of the cavity. Since the nodal quadrature rule employed is not exact in this case, a certain mass flow rate still remains.

Finally, Figure 8 shows details of the velocity module in the centre of the cavity. In the region of overlapping, the mass flow rate constraining method gives the best results. It should be also pointed out that the convergence of the problem is not affected by the constraining.

The mass conservation was illustrated for a simple flow because it enables to estimate clearly the effects of the constraining. However, the problem of incompatibility of the data is very likely to occur when using a Chimera method because it uses a Dirichlet transmission condition for the interface of patch meshes with the background mesh.

3.5. Example 3: force conservation

We will now present a strategy to conserve the components of the force (or traction) acting on the interface; see Figure 9. We propose to update the stress σ_i from the known stress σ_j

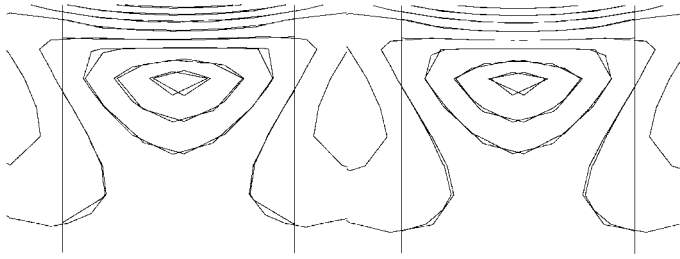


Figure 8. Example 2: detail of the velocity module contours of the two subdomains. (Left) Classical interpolation. (Right) Constrained interpolation.

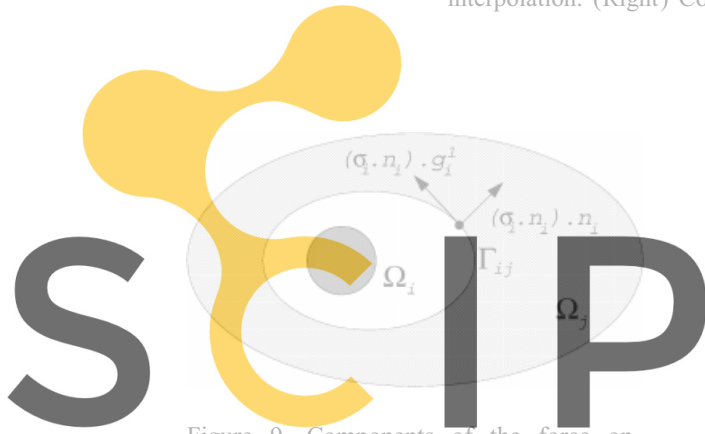


Figure 9. Components of the force on the interface.

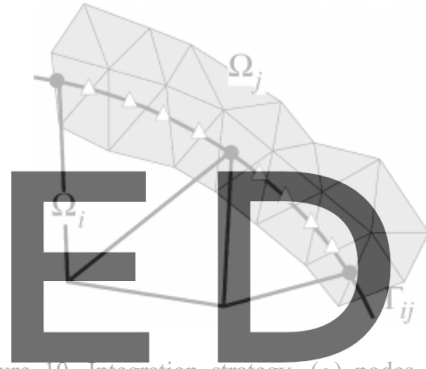


Figure 10. Integration strategy. (•) nodes of Γ_{ij} , (Δ) integration points.

Register for free at <https://www.scipedia.com> to download the version without the watermark

by solving the following problem in two dimensions:

$$\begin{aligned} & \text{minimize} \quad \int_{\Gamma_{ij}} |\boldsymbol{\sigma}_i \cdot \mathbf{n}_i - I^{ij}(\boldsymbol{\sigma}_j \cdot \mathbf{n}_i)|^2 d\Gamma \\ & \text{under the constraints} \\ & \int_{\Gamma_{ij}} (\boldsymbol{\sigma}_i \cdot \mathbf{n}_i) \cdot \mathbf{n}_i d\Gamma = \int_{\Gamma_{ij}} (\boldsymbol{\sigma}_j \cdot \mathbf{n}_i) \cdot \mathbf{n}_i d\Gamma \\ & \int_{\Gamma_{ij}} (\boldsymbol{\sigma}_i \cdot \mathbf{n}_i) \cdot \mathbf{g}_i^1 d\Gamma = \int_{\Gamma_{ij}} (\boldsymbol{\sigma}_j \cdot \mathbf{n}_i) \cdot \mathbf{g}_i^1 d\Gamma \end{aligned}$$

By introducing two Lagrange multipliers for the constraints, this problem leads to the solution of a matrix system of the form

$$\begin{bmatrix} \mathbf{M} & -\mathbf{R} & -\mathbf{S} \\ \mathbf{R}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{S}^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{t}_i \\ \mu_R \\ \mu_S \end{bmatrix} = \begin{bmatrix} \mathbf{M} \mathbf{I}^{ij}(\mathbf{t}_j) \\ r \\ s \end{bmatrix}$$

where \mathbf{t} is the nodal vector of the traction components, \mathbf{R} and r are the vector and scalar representing the normal force constraint, and \mathbf{S} and s are the vector and scalar representing

the tangential force constraint. The calculations of \mathbf{R} and \mathbf{S} are straightforward. However, the success of the conservation stems from calculating accurately the total force contribution of the adjacent subdomain j . In the example discussed previously, the constraint depended only on the solution in subdomain i . In the present problem, the force is known from j and if the mesh of i is too coarse with respect to the mesh in j , a special integration strategy has to be found to integrate r and s . In order to take into account the possible loss of conservation, the calculation of r and s will be performed by injecting a sufficient number of integration points on Γ_{ij} . The strategy is illustrated in Figure 10.

Note that an efficient element search strategy is therefore necessary in order to find a host element for each of these integration points. For example, the number of integration points to be chosen by element boundary could be related to the ratio of the local density of nodes of i to that of j .

4. CONCLUSION

We have developed a method for constraining the interface boundary conditions, in the framework of domain decomposition methods. As a first application, we have constrained Dirichlet interfaces with the zero mass flow rate equation and good results have been obtained. The method is general and can be applied to the conservation of any quantity involving the variable of the transmission condition.

REFERENCES

1. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 2001 in press.
2. Hughes TJR. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**:387–401.
3. Houzeaux G, Codina R. A domain decomposition method for the solution of moving subdomains in fluid dynamics. In *Proceedings of the 4th World Congress on Computational Mechanics*, Buenos Aires, Argentina, 1998.
4. Quarteroni A, Valli A. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications: Oxford, 1999.
5. Le Tallec P. Domain decomposition methods in computational mechanics. In *Computational Mechanics Advances*, vol. 1(2), Tinsley Oden J. (ed.). North-Holland: Amsterdam, 1994; 121–220.
6. Marini LD, Quarteroni A. An iterative procedure for domain decomposition methods: a finite element approach. In *1st International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Glowinski R, Golub GH, Meurant GA, Périaux J (eds). SIAM: Philadelphia, PA, 1988; 129–143.
7. Lions P-L. On the Schwarz alternating method I. *1st International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Glowinski R, Golub GH, Meurant GA, Périaux J (eds). SIAM: Philadelphia, PA, 1988; 1–42.
8. Lube G, Auge A, Otto FC, Kapurkin A. Domain decomposition methods for advection-dominated problems. In *Proceedings of the 3rd ECCOMAS Computational Fluid Dynamics Conference*, Paris, France, Wiley: New York, 1996; 1059–1065.
9. Lube G, Müller L, Otto F-C. A non-overlapping DDM of Robin–Robin Type for parabolic equations. In *11th International Conference of Domain Decomposition Methods*. ddm.org, Lai C-H, Bjørstad PE, Cross M, Widlund O (eds). 1999.
10. Trotta L. Multidomain finite elements for advection-diffusion equations. *Matematica* 1995; **474**:1–27.
11. Gastaldi F, Gastaldi L, Quarteroni A. ADN and ARN domain decomposition methods for advection-diffusion equations. In *9th International Conference of Domain Decomposition Methods*. ddm.org, 1998. Bjørstad PE, Espedal M, Keyes D (eds). Proceedings from the 9th International Conference, June 1996, Bergen, Norway.
12. Smith B, Bjørstad P, Gropp W. *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press: Cambridge, 1996.

13. Bernardi C, Maday Y, Patera AT. A new non conforming approach to domain decomposition: the mortar element method, In *Collège de France Seminar*. Brezis H, Lions J-L (eds). Pitman: Paris, 1994.
14. Farhat C, Roux FX. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering* 1991; **32**:1205–1227.
15. Marini LD, Quarteroni A. A relaxation procedure for domain decomposition methods using finite elements. *Numerische Mathematik* 1989; **55**:575–598.
16. Cebal JR, Löhner R. Conservative load projection and tracking for fluid-structure problems. Technical Report 96-0797, AIAA, 1996.